NAG Toolbox

nag_contab_tabulate_percentile (g11bb)

1 Purpose

nag_contab_tabulate_percentile (g11bb) computes a table from a set of classification factors using a given percentile or quantile, for example the median.

2 Syntax

```
[table, ncells, ndim, idim, icount, ifail] = nag_contab_tabulate_percentile(typ, isf, lfac, ifac, percnt, y, maxt, 'n', n, 'nfac', nfac, 'wt', wt)
[table, ncells, ndim, idim, icount, ifail] = g11bb(typ, isf, lfac, ifac, percnt, y, maxt, 'n', n, 'nfac', nfac, 'wt', wt)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 24: *weight* was removed from the interface; **wt** was made optional.

3 Description

A dataset may include both classification variables and general variables. The classification variables, known as factors, take a small number of values known as levels. For example, the factor sex would have the levels male and female. These can be coded as 1 and 2 respectively. Given several factors, a multi-way table can be constructed such that each cell of the table represents one level from each factor. For example, the two factors sex and habitat, habitat having three levels (inner-city, suburban and rural) define the 2×3 contingency table

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male			
Female			

For each cell statistics can be computed. If a third variable in the dataset was age then for each cell the median age could be computed:

Sex	**Habitat**		
	Inner-city	Suburban	Rural
Male	24	31	37
Female	21.5	28.5	33

That is, the median age for all observations for males living in rural areas is 37, the median being the 50% quantile. Other quantiles can also be computed: the p percent quantile or percentile, q_p, is the estimate of the value such that p percent of observations are less than q_p. This is calculated in two different ways depending on whether the tabulated variable is continuous or discrete. Let there be m values in a cell and let $y_{(1)}, y_{(2)}, \ldots, y_{(m)}$ be the values for that cell sorted into ascending order. Also,

associated with each value there is a weight, $w_{(1)}$, $w_{(2)}, \ldots, w_{(m)}$, which could represent the observed frequency for that value, with $W_j = \sum_{i=1}^{j} w_{(i)}$ and $W'_j = \sum_{i=1}^{j} w_{(i)} - \frac{1}{2}w_{(j)}$. For the p percentile let $p_w = (p/100)W_m$ and $p'_w = (p/100)W'_m$, then the percentiles for the two cases are as given below.

If the variable is discrete, that is, it takes only a limited number of (usually integer) values, then the percentile is defined as

$$\begin{cases} y_{(j)} & \text{if } W_{j-1} < p_W < W_j \\ \dfrac{y_{(j+1)} + y_{(j)}}{2} & \text{if } p_w = W_j. \end{cases}$$

If the data is continuous then the quantiles are estimated by linear interpolation.

$$\begin{cases} y_{(1)} & \text{if } p'_w \leq W'_1 \\ (1-f)y_{(j-1)} + fy_{(j)} & \text{if } W'_{j-1} < p'_w \leq W'_j \\ y_{(m)} & \text{if } p'_w > W'_m, \end{cases}$$

where $f = \left(p'_w - W'_{j-1}\right) / \left(W'_j - W'_{j-1}\right)$.

4 References

John J A and Quenouille M H (1977) *Experiments: Design and Analysis* Griffin

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

5 Parameters

5.1 Compulsory Input Parameters

1: **typ** – CHARACTER(1)

Indicates if the variable to be tabulated is discrete or continuous.

typ = 'D'
The percentiles are computed for a discrete variable.

typ = 'C'
The percentiles are computed for a continuous variable using linear interpolation.

Constraint: **typ** = 'D' or 'C'.

2: **isf**(**nfac**) – INTEGER array

Indicates which factors in **ifac** are to be used in the tabulation.

If **isf**$(i) > 0$ the ith factor in **ifac** is included in the tabulation.

Note that if **isf**$(i) \leq 0$, for $i = 1, 2, \ldots,$ **nfac** then the statistic for the whole sample is calculated and returned in a 1×1 table.

3: **lfac**(**nfac**) – INTEGER array

The number of levels of the classifying factors in **ifac**.

Constraint: if **isf**$(i) > 0$, **lfac**$(i) \geq 2$, for $i = 1, 2, \ldots,$ **nfac**.

4: **ifac**(ldf, **nfac**) – INTEGER array

ldf, the first dimension of the array, must satisfy the constraint $ldf \geq$ **n**.

The **nfac** coded classification factors for the **n** observations.

Constraint: $1 \leq \mathbf{ifac}(i,j) \leq \mathbf{lfac}(j)$, for $i = 1,2,\ldots,\mathbf{n}$ and $j = 1,2,\ldots,\mathbf{nfac}$.

5: **percnt** – REAL (KIND=nag_wp)

p, the percentile to be tabulated.

Constraint: $0.0 < p < 100.0$.

6: **y**(**n**) – REAL (KIND=nag_wp) array

The variable to be tabulated.

7: **maxt** – INTEGER

The maximum size of the table to be computed.

Constraint: **maxt** $\geq$ product of the levels of the factors included in the tabulation.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **y** and the first dimension of the array **ifac**. (An error is raised if these dimensions are not equal.)

The number of observations.

Constraint: **n** ≥ 2.

2: **nfac** – INTEGER

Default: the dimension of the arrays **isf**, **lfac** and the second dimension of the array **ifac**. (An error is raised if these dimensions are not equal.)

The number of classifying factors in **ifac**.

Constraint: **nfac** ≥ 1.

3: **wt**(:) – REAL (KIND=nag_wp) array

The dimension of the array **wt** must be at least **n** if $weight =$ 'W', and at least 1 otherwise

If $weight =$ 'W', **wt** must contain the **n** weights. Otherwise **wt** is not referenced.

Constraint: if $weight =$ 'W', $\mathbf{wt}(i) \geq 0.0$, for $i = 1,2,\ldots,\mathbf{n}$.

5.3 Output Parameters

1: **table**(**maxt**) – REAL (KIND=nag_wp) array

The computed table. The **ncells** cells of the table are stored so that for any two factors the index relating to the factor occurring later in **lfac** and **ifac** changes faster. For further details see Section 9.

2: **ncells** – INTEGER

The number of cells in the table.

3: **ndim** – INTEGER

The number of factors defining the table.

4: **idim**(**nfac**) – INTEGER array

The first **ndim** elements contain the number of levels for the factors defining the table.

5: **icount**(**maxt**) – INTEGER array

A table containing the number of observations contributing to each cell of the table, stored identically to **table**.

6: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **n** < 2,
or **nfac** < 1,
or $ldf <$ **n**,
or **typ** $\neq$ 'D' or 'C',
or $weight \neq$ 'U' or 'W',
or **percnt** ≤ 0.0,
or **percnt** ≥ 100.0.

ifail $= 2$

On entry, **isf**$(i) > 0$ and **lfac**$(i) \leq 1$, for some i,
or **ifac**$(i, j) < 1$, for some i, j,
or **ifac**$(i, j) >$ **lfac**(j), for some i, j,
or **maxt** is too small,
or $weight =$ 'W' and **wt**$(i) < 0.0$, for some i.

ifail $= 3$

At least one cell is empty.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

The tables created by nag_contab_tabulate_percentile (g11bb) and stored in **table** and **icount** are stored in the following way. Let there be n factors defining the table with factor k having l_k levels, then the cell defined by the levels i_1, $i_2, \ldots, i_n$ of the factors is stored in the mth cell given by:

$$m = 1 + \sum_{k=1}^{n} [(i_k - 1)c_k],$$

where $c_j = \prod_{k=j+1}^{n} l_k$, for $j = 1, 2, \ldots, n-1$ and $c_n = 1$.

9 Example

The data, given by John and Quenouille (1977), is for a 3×6 factorial experiment in 3 blocks of 18 units. The data is input in the order, blocks, factor with 3 levels, factor with 6 levels, yield, and the 3×6 table of treatment medians for yield over blocks is computed and printed.

9.1 Program Text

```
     function g11bb_example

fprintf('g11bb example results\n\n');

ifac = [nag_int(1),1,1; 1,2,1; 1,3,1; 1,1,2; 1,2,2; 1,3,2;
                 1,1,3; 1,2,3; 1,3,3; 1,1,4; 1,2,4; 1,3,4;
                 1,1,5; 1,2,5; 1,3,5; 1,1,6; 1,2,6; 1,3,6;
                 2,1,1; 2,2,1; 2,3,1; 2,1,2; 2,2,2; 2,3,2;
                 2,1,3; 2,2,3; 2,3,3; 2,1,4; 2,2,4; 2,3,4;
                 2,1,5; 2,2,5; 2,3,5; 2,1,6; 2,2,6; 2,3,6;
                 3,1,1; 3,2,1; 3,3,1; 3,1,2; 3,2,2; 3,3,2;
                 3,1,3; 3,2,3; 3,3,3; 3,1,4; 3,2,4; 3,3,4;
                 3,1,5; 3,2,5; 3,3,5; 3,1,6; 3,2,6; 3,3,6];
y   = [            274;   361;   253;   325;   317;   339;
                   326;   402;   336;   379;   345;   361;
                   352;   334;   318;   339;   393;   358;
                   350;   340;   203;   397;   356;   298;
                   382;   376;   355;   418;   387;   379;
                   432;   339;   293;   322;   417;   342;
                    82;   297;   133;   306;   352;   361;
                   220;   333;   270;   388;   379;   274;
                   336;   307;   266;   389;   333;   353];

lfac   = [nag_int(3); 3; 6];
isf    = [nag_int(0); 1; 1];
maxt   = prod(lfac(isf~=0));
maxt   = nag_int(maxt);

typ    = 'C';
percnt = 50;

% Compute classification table
[table, ncells, ndim, idim, icount, ifail] = ...
  g11bb( ...
         typ, isf, lfac, ifac, percnt, y, maxt);

% Display results
fprintf(' Table for %4dth percentile\n\n', percnt);
ncol = idim(ndim);
nrow = ncells/ncol;
table  = transpose(reshape(table,[ncol,nrow]));
icount = transpose(reshape(icount,[ncol,nrow]));
for i = 1:nrow
  row = [table(i,:); double(icount(i,:))];
  fprintf('%8.2f(%2d)', row);
  fprintf('\n');
end
```

9.2 Program Results

```
     g11bb example results

 Table for   50th percentile

  226.00( 3)  320.25( 3)  299.50( 3)  385.75( 3)  348.00( 3)  334.75( 3)
  329.25( 3)  343.25( 3)  365.25( 3)  370.50( 3)  327.25( 3)  378.00( 3)
  185.50( 3)  328.75( 3)  319.50( 3)  339.25( 3)  286.25( 3)  350.25( 3)
```